SpanLuke: Enhancing Legal NER using SpanMarkers and LoRA

Davide Buoso[1] **Enrico Capuano**[1] **Andrea Caselli**[1] **Destiny Okpekpe**[1]

[1]*Politecnico di Torino*, Turin, Italy, {name.surname}@studenti.polito.it

Abstract—**Legal Named Entity Recognition is a focal point in NLP systems in legal domain, due to its potential to streamline processes and enhance decision-making accuracy. This paper delves into the SpanMarker technique for span-level representation of entity, the LUKE model for enhanced entity recognition and LoRA for efficient fine-tuning of large models. The study evaluates these methodologies, individually and in synergy, to improve the accuracy and performance of legal NLP systems. Additionally, a new dataset (EDGAR-NER) has been explored. In-depth experimentation reveals the potential of these approaches. This research contributes to ongoing efforts in leveraging NLP to enhance legal text process, reducing the time needed to train models to achieve this goal. The code is available at: `https://github.com/lambdavi/SpanLuke`.**

Index Terms—**Named Entity Recognition, Legal NER, SpanMarker, Luke, E-NER, PEFT, LoRA**

I. Problem statement

The integration of Natural Language Processing (NLP) techniques within the legal domain has revolutionized the landscape of legal research, content retrieval, and decision-making processes. Key tasks in this domain include Legal Named Entity Recognition, which have garnered considerable attention due to their implications for streamlining legal processes and enhancing decision-making accuracy. Given an annotated input sentence, taken from a legal corpora, the goal of the task is to identify the entities contained in it. Our study focuses on leveraging innovative methodologies such as the SpanMarker [1] technique for precise entity recognition through the use of span representations, the LUKE [2] model for domain-specific entity understanding, and the LoRA [3] (Low Rank Adaptation of LLMs) for the fine-tuning of predictive models. We expect, using this framework, a general improvement of the scores and when using the LoRA technique, a reduction in training time and in particular the number of parameters used. Through rigorous experimentation and analysis, we evaluate the efficacy of these enhancements in improving the accuracy and we prove the validity of the previous hypothesis. Our experiments were performed on the dataset introduced in [4] (we refer to this task/dataset with ”L-NER”). We also briefly explore the augmentation of this dataset, to increase the number of entries of the dataset, in order to subsequently improve the accuracy of the system. However, we understand it is not an easy task, given the specificity of the vocabulary used. We investigate the integration of these techniques into a holistic framework, showcasing their synergistic effects in enhancing the task. Our findings shed light on the potential of these state-of-the-art approaches to revolutionize legal text processing, paving the way for more accurate and efficient decision-making processes in the legal domain.

Summing up, our main contributions are:

- Adaptation of SpanLuke: combination of Luke model with span-level representation.
- Analysis of LoRA method for faster and efficient fine-tuning.
- Exploration of a new dataset EDGAR-NER and a tentative of data augmentation on the Legal NER dataset.

II. Methodology

Recent advancements in NLP, particularly the utilization of transformer-based models, have paved the way for more accurate solutions to a wide rage tasks. In the NER scenario, a further enhancement, which inspired our work and laid the base of the experiments, can be found in the LUKE model [2], a transformer that exploits an entity-aware self-attention mechanism: it allows entities to be tokens themselves, and not only words. Techniques such as entity-aware attention mechanisms and domain-specific pre-training have shown promise in improving the performance of L-NER models [5], enabling more precise identification of legal entities within textual data. Being in the legal sector though, it is imperative to achieve the greatest possible accuracy: due to the high stakes involved in legal proceedings, even minor inaccuracies or wrong evaluations can have significant consequences for clients, judicial decisions, and the administration of justice.

The dataset used for this analysis (L-NER) is sourced from Indian court judgments spanning from 1950 to 2021. Representative samples from judgments between 1950 and 2017 were annotated for training data, while judgments from 2017 to 2021 were utilized for selecting test data. The dataset comprises 9435 judgment sentences and 1560 preambles, each separately provided due to differences in entity representation between the preamble and judgment. The annotation is focused on individual sentences for NER, without considering document-level context, prioritizing flat entity annotations to distinguish entities like ”Bank of China” as ORG while excluding internal entities such as ”China” as GPE within that entity.

A. SpanLuke

In the literature, different span-level models have been proposed, such as solid markers [6] or Packed-Levitated Marker

[7]. In particular, the most promising technique is the latter, which is based on the idea to pack levitated marker pairs together for all spans within a text. The main drawback of this approach is that BERT-style encoders suffer from quadratic complexity. T. Arsen [1] introduces a new piece of software called SpanMarker, which tries to mitigate the main limitations of PL-Markers. To address it, SpanMarker introduces a restriction requiring each token embedding to be used in only one span embedding. This is achieved by introducing a pair of special ”<start>” and ”<end>” marker tokens for each individual span. The start marker token serves to observe the first token within the span, while the end marker token observes the last token of the last word in the span. By employing this approach, the resulting span embeddings are not composed of the embeddings of the text tokens themselves, but rather the embeddings of the observing start and end markers. Importantly, these span embeddings offer improved classification capabilities. This is because the marker embeddings uniquely correspond to a single span with a single label, as opposed to being associated with multiple spans that may have different labels. SpanMarker considers all possible spans (or n-grams) of words up until a predefined maximum entity length, for example up to 8 words. A span is considered valid if it does not contain more words than the maximum entity length allows. An example (subset) of the spans created from a sentence is presented in Figure 1, while the picture representing how the SpanMarker model works is presented in Figure 2.

```
 1  Andorra
 2  Andorra is
 3  Andorra is located
 4  Andorra is located between
 5  Andorra is located between France
 6  Andorra is located between France and
 7  Andorra is located between France and Spain
 8  Andorra is located between France and Spain.
 9  is
10  is located
```

Fig. 1. Span Example of the sentence: ”Andorra is located between France and Spain”

In this work we combine the entity attention mechanism of LUKE [2] with the power of SpanMarker. We name this combination SpanLUKE. In our scenario we focus on legal text, in particular the identification of legal entities contained in it. The entities in this domain are often composed of multiple words, hence the use of span embeddings could be highly beneficial. Also, the use of LUKE can enhance the accuracy of the identification thanks to its modified attention, which is different for words and entities and their combination. We demonstrate how this approach brings to higher results with respect to the baseline.

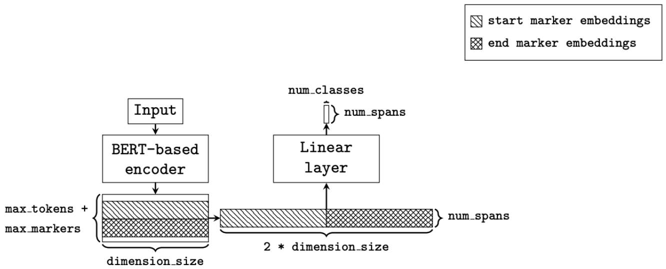

Fig. 2. Span Architecture: The embeddings of start and end markers are combined and fed through a linear layer, producing logits of shape: (number of spans, number of classes).

B. LoRA

In the field of Natural Language Processing, but generally in Deep Learning, it’s common practice to utilize large base pre-trained models, such as the Bert’s and the GPT’s family, which are fine-tuned for a specific downstream task and domain-specific dataset. However, this approach needs a full-tuning of all the parameters of the original model, which can be a critical challenge with large models, such as GPT-3 [8] with 175 Billions parameters. Previous work tried to optimize fine-tuning by adding adapter layers ([9]–[12]) and optimizing the input layers ([13]–[16]).

However, both of them had major downside: the first approach introduces inference latency, since computation through adapter layers have to be processed sequentially, while the second approach reduces the sequence length available to process downstream tasks.

In this context LoRA, introduced by E. J. Hu et al. [3], finds its place. Following previous works stating that the learned over-parametrized models reside on a low intrinsic dimension, the authors hypothesize that also the change in weights during model adaptation has low “intrinsic rank”. So by updating just a fraction of all the weights we get most of the benefits of fine-tuning.

The main idea is to represent the weight matrix W_0 with a low-rank decomposition $W_0 + BA$, where the first component is frozen and only the low-rank matrix BA is updated with gradient. This approach has gained us improvements in our task: reduced the number of weights to update, thus shortened the time needed for fine-tuning and let the model focus on meaningful updates. Lora working mechanism is depicted in Figure 3.

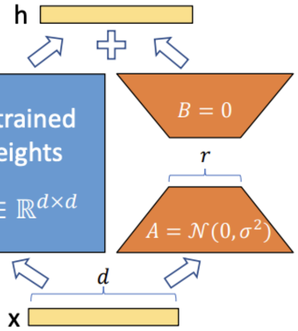
h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Fig. 3. Lora inner working

TABLE I
Results on Legal NER dataset: Results of original paper and baseline are presented in the first rows. The other rows are the performance obtained by Luke-Lora, SpanLuke, SpanLuke+ and the SpanLuke-Lora

Description	**F1 Type-Match**	**F1 Partial**	**F1 Exact**	**F1 Strict**	**Train. Params (M)**	**Time (s)**
Luke BASE (Paper)	93.49	92.73	89.85	88.89	278	-
Luke LARGE (Paper)	**94.20**	93.45	90.68	89.88	562	-
Luke BASE Baseline	92.97	91.97	88.91	87.94	278	4171
Luke BASE Baseline+	92.47	91.44	88.25	87.36	278	4171
Luke LoRA+ (Ours)	92.51	90.82	87.07	86.28	6	**3100**
Span-Luke (Ours)	92.32	93.07	91.33	89.83	280	4091
Span-Luke+ (Ours)	92.77	**93.65**	**92.37**	**91.0**	280	4091
Span-Luke-LoRA+ (Ours)	92.75	93.44	92.09	90.67	7	3681

C. Dataset Exploration and Augmentation

In this work we also explored a relatively new dataset EDGAR-NER (E-NER) introduced in [17]. This dataset is based on US legal company filings, extracted from US Securities and Exchange Commission's EDGAR dataset.

The dataset refers to different filings, collected from the year 2010 and divided in 52 documents. It is composed by 7 entity types: *Location*, *Person*, *Business*, *Government*, *Court*, *Legislation/Act* and *Miscellaneous*. We also provide two jsonl containing the annotated filings, with tokens and NER tags on each line. One of the main challenge of this dataset is the scarcity and quality of data that could lead to both overfitting and training instability. If we compare E-NER to a famous NER dataset, such as ConLL-2003 [18], we can notice significant differences in both the number of sentences (11.696 vs 22,136) and NE phrases (8,821 vs 35,088). In section III we explain the analysis done on this dataset.

Another point we wanted to explore was the possibility to enhance the data available. In fact, one of the main challenges of the task is the peculiar vocabulary used, very precise and with many uncommon words with respect to the ones used for most of the pre-training in base models. For this reason our group thought to leverage augmentation in order to have a larger dataset for the fine-tuning process.

As shown in Computer Vision, by injecting various type of noise to images, we can feed to models more data and obtain more accurate results and enhance generalization. Such strategy is trending also in NLP and with swaps, synonymous and deletion we modified our training sentences and merge them to the original ones, doubling the size of the training dataset.

However this method did not achieve the desired outcome, as we suspected, resulting in a significant decrease in performance (roughly 10%).

By watching attention scores, accuracy on certain entity and some augmented sentences we try to justify this behaviour with two hypothesis:

- the entities are recognized thanks to particular tokens and any change to those deprive the models from the most important knowledge to make predictions
- the search of synonyms is not trivial in the legal context.

III. EXPERIMENTS

We started the experiments considering as baseline the one introduced in [5]. Due to the limited hardware availability we did not reach the exact results obtained in the original paper, since a batch size of 256 were not supported by our technological means. For the experiments, we used Kaggle as substitute of Google Colab, offering a single Tesla P100. For sake of reproducibility in the code has been set a seed (42) for all processes regarding PyTorch. This should make the whole process as deterministic as possible (with all due limitations). As tokenizer, for all the experiments we used RobertaTokenizerFast. We investigated the use of SpanMarker to prove its benefits against the baseline. For what concerns the baseline, we mantained most of the original hyperparameters of [5], changing the batch size from 256 to 8. Furthermore, due to limitation of time and hardware, we analyzed only the base version of the LUKE model, since the improvements with respect to the large model were not substantial, considering also the limited size of the training dataset. The learning rate is set at $1e-4$, expect in the case of Lora Models, where we found that an higher learning rate performed better ($1e-3$). For the LoRA hyperparameters we chose Lora Rank = 32 and Lora Alpha = 16. As for the layers to apply the technique to, we explored two options: apply it to query and value layers (as in [3]) and apply it to all linear layers (as in [19]). We noticed that the first choice yielded to faster training but worse performance while the second one (that has been chosen for the experiments) registered some benefits on the training time but still great performance.

We then explored the use of Gradient Accumulation. It works by accumulating gradients over several batches, and only stepping the optimizer after a certain number of batches have been processed. If this technique has been used, the row of the results table has been marked with a '+'. We set the accumulation steps to 2.

As for the SpanLuke model, we trained it setting the maximum number of words per entity to 6. This is justified by an analysis performed on the dataset. We discovered that the average number of words per entity, in the training set, was approximately 2.94 ± 2.6. The results obtained on the Dev split are reported in Table I. Additionally, the best performing model is published on HuggingFace platform [1] for reproducibility and future works.

We briefly tested the benefits obtained by the augmentation. We doubled the number of entries of the dataset. However, we knew in advance that the legal vocabulary is highly domain-specific and trying to augment the dataset would have not been trivial. The results reflected these expectations and we don't think presenting these results would be useful.

We then analyzed the EDGAR-NER dataset, a relatively new dataset. The goal of this analysis was to test if the pre-training on the Legal NER dataset could be beneficial for future works in similar legal scenarios and, at the same time, demonstrate the robustness of the SpanLuke model.

We explored four different scenarios: Luke and SpanLuke finetuned directly on E-NER, and the same models pretrained on Legal-NER and subsequently finetuned on E-NER.

Unfortunately, the authors of [17] did not specify, neither gave an official split for the evaluation of their dataset. Therefore, we had to provide a new (un)official split, in order to allow both to reproduce the results and to test new models against our new baseline. The split was obtained dividing the document in sentences and, after shuffling, we divided the train and test split using a test size of 20%. We provide in the repository the train and test files in jsonl format.

Following this reasoning, the baseline was obtained based using BERT base (as in [20]). For this task, the models were trained over 5 epochs with a weight decay of 0.01 and a warmup ratio of 0.06, as in previous experiments. After observing that in this dataset the number of words for entity was similar to Legal-NER, we mantained the corresponding SpanMarker hyperparameter at 6. The other hyperparameters and results are presented in Table II.

Model	**BS**	**LR**	**F1**
BERT (E)	8	1e-4	92.95
Luke (E)	8	1e-4	93.74
Span-Luke (E)	8	1e-4	93.86
Luke (L− >E)	8	7.5e-5	93.31
Span-Luke (L− >E)	8	7.5e-5	**95.92**

TABLE II
Finetuning on EDGAR-NER: where **E** stands for E-NER and **L** stands for Legal NER

IV. Analysis

Through the combined use of SpanMarker and LUKE, as one can observe from Table I, we obtained remarkable results, surpassing even the baseline set with Luke-Large, on most of the metrics. We also observed than using, the training with the Levitated Markers takes even less time than compared to the base model, while bringing considerable improvements in term of performance on the F1 Partial, F1 Exact and F1 Strict metrics (introduced in [21]). With the use of LoRA, we obtained a relevant speed up (40%) with respect to Luke Base, but with a trade-off on performance. As for the comparison between vanilla Span-Luke, and SpanLuke+Lora, we observed a reduction in training time almost of 9%. The gains, in this last scenario, are not very significant, considering also the fluctuation of the training time, due to the cloud computation. We believe this is due to the fact that the additional time is needed by the SpanMarker framework to compute the span mechanism, which is not optimized by the inner working of LoRA. Theoretically, LoRA could deliver the same performance on very Large Language Models, but in our experiments, we noticed a slight reduction of F1 score, still mantaining remarkable results. We think this is reasonable and also impressive considering the number of parameters trained (reduction of about 97%). In Table II, we demonstrated how SpanLuke is also able to adapt to the EDGAR-NER dataset effectively, without the need of a pre-training on similar datasets. Also, as expected, the SpanLuke model pre-trained on Legal NER is significantly better than finetuning it directly. Luke pre-trained delivers worse performance with respect to the same model directly fine-tuned on E-NER. Our hypothesis is that the model needed more time to "adapt" to the new vocabulary, since L-NER is more focused on legal domain and E-NER is a mix of the financial and the legal ones. We noticed the loss was still decreasing, so probably, on longer runs (e.g. on 10 epochs) this model would have delivered better results.

V. Conclusions

In our work, we demonstrate how these techniques can be applied to effectively tackle datasets in the legal domain. We tested all these new approaches on Luke-Base and we observed remarkable performance, even against the baselines of larger models. We believe that these techniques can be shifted to bigger models effortlessly and with further improvements. This could be the first direction of future works. Especially LoRA is proved to deliver better results on very big models. Additionally, the exploration of more advanced PEFT techniques (e.g. QLoRA and AdaLora), could further improve the gains in terms of speed and efficiency. Our tentative to augment the dataset demonstrated how "standard" augmentation techniques are not easily applicable to the legal domain, which requires more advanced techniques, such as generative models. In fact, the augmentation should be tailored for this peculiar domain, preserving the entities while changing the other words of the sentence with synonyms coherent with the context. In addition to those mentioned above, other ideas which could represent promising approaches for future works, are:

- Use of open-source LLM as LLama/Mistral, specifically finetuned with LoRA for NER.
- Pre-train encoders on large legal corpora to efficiently learn the semantic and syntactic relations among words, with the goal to further finetuning the models on the L-NER task.

[1] https://huggingface.co/lambdavi/span-marker-luke-legal

References

[1] T. Aarsen, "SpanMarker for Named Entity Recognition." [Online]. Available: https://github.com/tomaarsen/SpanMarkerNER

[2] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, "Luke: Deep contextualized entity representations with entity-aware self-attention," 2020.

[3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[4] P. Kalamkar, A. Agarwal, A. Tiwari, S. Gupta, S. Karn, and V. Raghavan, "Named entity recognition in indian court judgments," 2022.

[5] I. Benedetto, A. Koudounas, L. Vaiani, E. Pastor, E. Baralis, L. Cagliero, and F. Tarasconi, "PoliToHFI at SemEval-2023 task 6: Leveraging entity-aware and hierarchical transformers for legal entity recognition and court judgment prediction," in *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*. Association for Computational Linguistics, 2023.

[6] L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," *CoRR*, vol. abs/1906.03158, 2019. [Online]. Available: http://arxiv.org/abs/1906.03158

[7] D. Ye, Y. Lin, P. Li, and M. Sun, "Packed levitated marker for entity and relation extraction," 2022.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[9] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.

[10] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Learning multiple visual domains with residual adapters," *Advances in neural information processing systems*, vol. 30, 2017.

[11] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych, "Adapterfusion: Non-destructive task composition for transfer learning," *ArXiv*, vol. abs/2005.00247, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218470208

[12] A. Rücklé, G. Geigle, M. Glockner, T. Beck, J. Pfeiffer, N. Reimers, and I. Gurevych, "Adapterdrop: On the efficiency of adapters in transformers," *arXiv preprint arXiv:2010.11918*, 2020.

[13] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," *arXiv preprint arXiv:2101.00190*, 2021.

[14] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," *arXiv preprint arXiv:2104.08691*, 2021.

[15] K. Hambardzumyan, H. Khachatrian, and J. May, "Warp: Word-level adversarial reprogramming," *arXiv preprint arXiv:2101.00121*, 2021.

[16] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "Gpt understands, too," *AI Open*, 2023.

[17] T. W. T. Au, V. Lampos, and I. Cox, "E-NER — an annotated named entity recognition corpus of legal text," in *Proceedings of the Natural Legal Language Processing Workshop 2022*, N. Aletras, I. Chalkidis, L. Barrett, C. Goanță, and D. Preoțiuc-Pietro, Eds. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 246–255. [Online]. Available: https://aclanthology.org/2022.nllp-1.22

[18] E. F. T. K. Sang and F. D. Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," 2003.

[19] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," 2023.

[20] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[21] I. Segura-Bedmar, P. Martínez, and M. Herrero-Zazo, "SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013)," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, S. Manandhar and D. Yuret, Eds. Atlanta, Georgia, USA: Association for Computational Linguistics, Jun. 2013, pp. 341–350. [Online]. Available: https://aclanthology.org/S13-2056